3.11. Magistrala PCI

3.11.1. Prezentare generala

În anul 1992, firma *Intel* a anuntat prima versiune (1.0) a unei specificatii pentru o magistrala locala, denumita PCI *(Peripheral Component Interconnect).* Intentia initiala era realizarea unui standard care sa permita interconectarea circuitelor rapide pe placa de baza, deoarece cu fiecare generatie de microprocesoare, *Intel* trebuia sa modifice arhitectura magistralei locale pentru a mari performantele sistemului, ceea ce implica si modificarea circuitelor de interfata pentru periferice.

Magistrala PCI urma sa se conecteze la magistrala locala a procesorului prin intermediul unui circuit special destinat acestui scop. Astfel, la fiecare schimbare a procesorului si a magistralei locale, trebuia schimbat doar circuitul de legatura, circuitele de interfata cu perifericele nefiind afectate.

Specificatia PCI initiala nu prevedea conectori de extensie. *Intel* a actualizat specificatiile PCI pentru ca aceasta sa admita si conectori de extensie. Astfel magistrala PCI a fost definita detaliat din punct de vedere electric si functional, ajungând cea mai utilizata arhitectura de magistrala. Versiunea 2.0 a aparut în 1993, iar versiunea 2.1 în 1995.

Sunt posibile transferuri de 32 sau de 64 biti. În versiunea 2.1, frecventa maxima a ceasului este de 66 MHz, ceea ce permite obtinerea unei rate maxime de tmnsfer de 528 MB/s.

Standardul de extensie a placii PCI defineste o familie de conectori pentru adaptoare de 5 V sau 3,3 V, cu variante de 32 si 64 de biti. Este de asteptat ca în viitor majoritatea conectorilor de extensie sa utilizeze surse de alimentare de 3,3 V in loc de 5 V.

Specificatiile PCI pennit utilizarea a doua din cele trei metode de conectare la magistrala locala a procesorului: conectarea printr-un buffer si conectarea de tip statie de lucru. Datorita avantajelor legate de performante si de flexibilitate, conectarea de tip statie de lucru este preferata. Se prezinta în Figura 3.20 schema bloc a unui sistem cu magistrala PCI, utilizind acest tip de conectare.

Magistrala PCI este conectata la magistrala sistem a procesorului prin intermediul unei punti UCP/PCI, având acces direct la memoria principala. În acest fel, transferurile între UCP si memoria cache, respectiv între dispozitivele de I/E si memoria principala pot avea loc simultan. Dispozitivele de I/E de viteza ridicata, ca a adaptoarele grafice si adaptoarele de retea, care solicita în masura redusa procesorul, sunt conectate direct la magistrala PCI. Dispozitivele care trebuie sa se conformeze altor standarde de magistrala., ca ISA sau SCSI, se interfateaza cu magistrala PCI printr-o punte PCI/ISA, respectiv un adaptor SCSI.

Un sistem poate fi realizat si fara utilizarea puntilor de legatura. În acest caz, toate componentele, inclusiv procesorul si memoria principala, se interfateaza direct cu magistrala PCI.

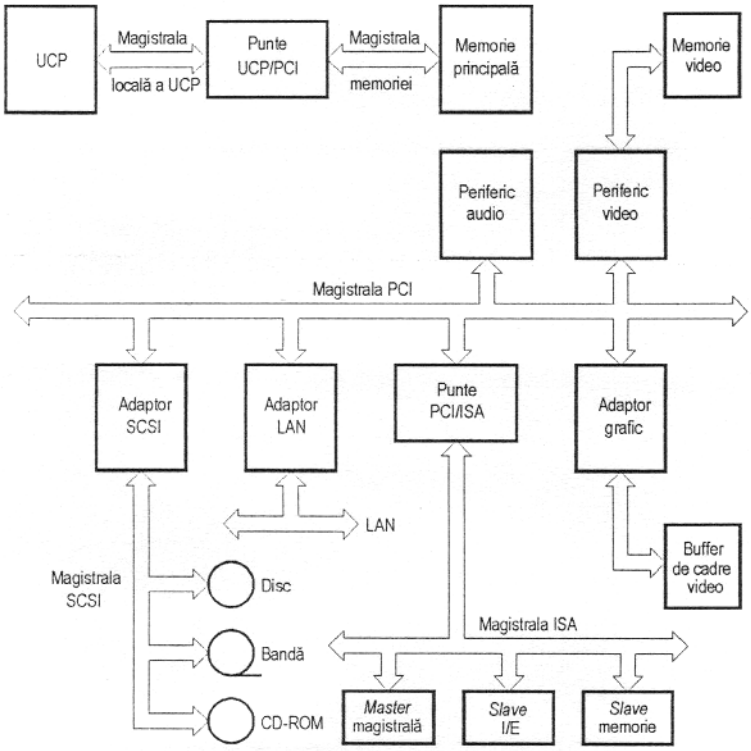

Figura 3.20. Schema bloc a unui sistem cu magistrală PCI

3.11.2. Semnalele magistralei PCI

Magistrala PCI este sincrona, la fel ca toate magistralele PC, inclusiv modelul original IBM PC. Toate tranzactiile pe magistrala PCI sunt între un modul master, denumit oficial initiator (initiator), si modul slave, denumit oficial tinta (target). Pentru a mentine redus numarul de pini PCI, liniile de adresa si de date sunt multiplexate. În acest mod, pentru placile PCI sunt necesari 64 pini, pentru semnalele de adresa plus cele de date, chiar daca PCI suporta adrese de 64 biti si date pe 64 de biti.

Pinii multiplexati de adrese si date functioneaza dupa cum urmeaza. La o operatie de citire, pe durata ciclului 1, master-ul pune adresa pe magistrala. În ciclul 2 master-ul preia adresa de pe magistrala, iar magistrala este oferita si poate fi utilizata de modulul slave. În ciclul 3, modul slave furnizeaza date curente. La operatia de scriere, magistrala nu trebuie eliberata, deoarece master-ul pune pe ea atât adresa cât si datele. Cu toate acestea, tranzactia minima are totusi trei cicluri. Daca modulul slave nu este capabil sa raspunda în timpul celor trei cicluri, el poate insera stari de asteptare (wait). Sunt permise, de asemenea, transferuri de blocuri de dimensiune nelimitata, precum si alte câteva tipuri de cicluri de magistrala.

Fiecare dispozitiv conectat la magistral a PCI poate functiona fie ca *slave*, fie ca *master/slave*. Un dispozitiv *master* se mai numeste *initiator*, iar unul *slave* se mai numese *tinta (target)*. Un dispozitiv tinta nu poate initia transferuri pe magistrala.

Pentru a reduce numarul pinilor si dimensiunea conectorilor necesari pentru unitatile compatibile PCI, adresele si datele sunt multiplexate, setul de linii pentru linii si date fiind notat cu *AD*. O tranzactie tipica pe magistrala implica doua faze. În prima faza se transmite o adresa pe liniile *AD*, iar în faza a doua se transmit una sau mai multe cuvinte de date pe aceleesi linii. Toate operatiile magistralei sunt sincronizate cu ajutorul unui semnal de ceas, deoarece magistrala este sincrona. Sunt prevazute însa semnale care permit inserarea starilor de asteptare de catre dispozitivele mai lente.

În Figura 3.21 se prezinta principalele semnale ale unui dispozitiv *master/slave*. În partea stânga se afla semnalele obligatorii, necesare pentru transferurile de baza utilizând cuvinte de maxim 32 de biti. În partea dreapta se afla semnalele optionale, care permit transferuri de date utilizand cuvinte de 64 biti, semnale care permit controlul întreruperilor si semnale pentru executarea altor functii mai putin uzuale.

Magistrala PCI are un numar de semnale obligatoriisi un numar de semnale optionale, prezentate în Fig.3-21. Restul de 120 sau 184 de pini sunt utilizati pentru alimentare, masa si diferite functii înrudite, care nu sunt listate aici.

Urmeaza o trecere în revista a semnalelor magistralei PCI. Vom începe cu semnalele obligatorii (32 de biti) si apoi vom trece la semnalele optionale (64 de biti).

[AD 31-0](Adress/Data)

Reprezinta magistrala multiplexata de adrese si de date. În timpul fazei de adrese, pe aceasta magistrala se transmite adresa de start a tranzactiei. În timpul fazei de date, pe liniile *AD[31-0]* se transmit date provenite de la initiator (la o scriere) sau de la tinta adresata (la o citire).

C/BE[3-0] (Command or Byte Enable)

În timpul fazei de adrese aceste linii definesc o comanda pe care inittiatorul o utilizeaza pentru a indica tipul tranzactiei necesare. Dintre comenzile posibile se amintesc citirea din memorie, scrierea in memorie, citirea de la un dispozitiv de I/E, scrierea la un dispozitiv de I/E, achitarea unei întreruperi etc. În timpul fazei de date aceste linii sunt utilizate de initiator pentru a indica octetii care trebuie transferati din cadrul cuvântului dublu adresat si grupele de linii ale magistralei *AD* care trebuie utilizate pentru transferul datelor.

PAR (Parity)

Reprezinta semnalul de paritate para pentru liniile *AD[31-0]* si *C/BE [3-0]*. Paritatea este generata de initiator dupa terminarea fazei de adrese si dupa terminarea fiecarei faze de date la o tranzactie de scriere. Paritatea este generata de tinta adresata dupa terminarea fiecarei faze de date la o tranzactie de citire.

FRAME (Cycle Frame)

Este activat de initiator si indica începutul si durata unei tranzactii pe magistrala. O tranzactie poate consta din una sau mai multe transferuri de date între initiatorul curent si tinta adresata. Semnalulul FRAME este dezactivat atunci când initiatorul este pregatit sa încheie faza finala de date.

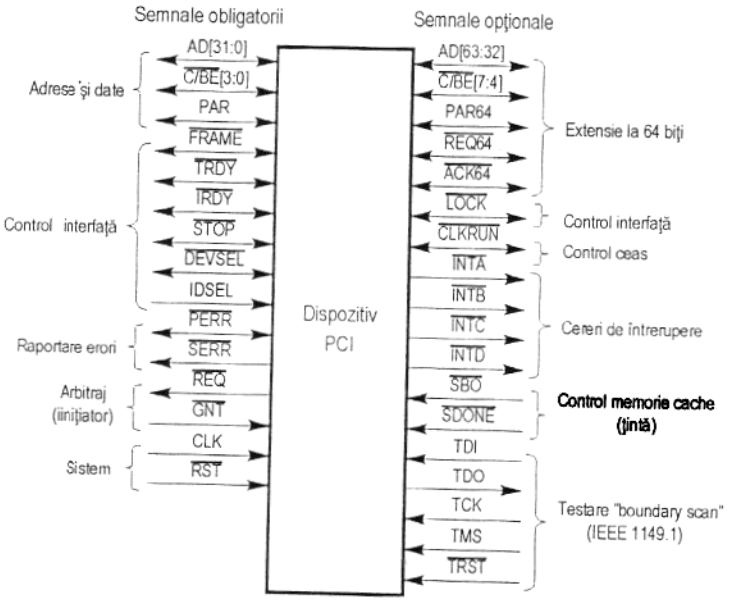

Figura 3.21. Semnalele unui dispozitiv PCI

TRDY (Target Ready)

Este activat de dispozitivul tinta adresat atunci când acest dispozitiv este pregatit pentru un transfer de date (poate executa faza curenta de date). Faza de date este terminata atunci când tinta activeaza *TRDY* si initiatorul activeaza *IRDY* la frontul crescator al semnalului de ceas. În timpul unei operatii de citire, semnalul *TRDY* activat indica faptul ca tinta a depus date valide pe magistrala de date. În timpul unei operatii de scriere, semnalul *TRDY* activat indica faptul ca tinta este pregatita sa accepte datele de la initiator. Sunt inserate stari de asteptare în faza curenta de date pâna când ambele semnale *TRDY* si *IRDY* sunt activate.

IRDY (Initiator Ready)

Este activat de initiatorul tranzactiei si semnaleaza momentul în care initiatorul este pregatit pentru un transfer de date. În timpul unei operatii de scriere, semnalul *IRDY* activat indica faptul ca initiatorul a depus date valide pe magistrala de date. În timpul unei operatii de citire, semnalul *IRDY* activat indica faptul ca initiatorul este pregatit sa accepte datele de la tinta adresata.

STOP

Prin activarea acestui semnal, dispozitivul tinta solicita initiatorului oprirea tranzactiei în curs in timpul fazei curente de date.

DEVSEL (Device Select)

Este activat de dispozitivul target pâna atunci când acesta si-a decodificat adresa. Daca un dispozitiv *master a* initiat un transfer si nu detecteaza semnalul *DEVSEL* activ în cursu1 a sase perioade de ceas, va presupune ca tinta nu poate raspunde sau ca adresa respectiva nu este utilizata. Va rezulta un abandon din partea dispozitivului *master.*

IDSEL (Initialization Device Select)

Este o intrare pentru un dispozitiv PCI si este utilizat pentru seletia circuitului in timpul accesului la unul din registrele de configurare ale dispozitivului.

PERR (Parity Error)

Indica o eroare de paritate. Generarea informatiei de paritate este obligatorie pentru toate dispozitivele PCI care transmit adrese si date pe magistrala *AD.* Detectarea si raportarea erorilor de paritate de catre dispozitivele PCI este necesara în general (în anumite cazuri, erorile de paritate pot fi ignorate). Daca se detecteaza o eroare de paritate de catre un dispozitiv tinta în timpul unei faze de date la o operatie de scriere, tinta trebuie sa activeze semnalul *PERR.* Tinta poate continua tranzactia sau poate activa semnalul *STOP* pentru terminarea prematura a tranzactiei. Daca se detecteaza o eroare de paritate de catre un dispozitiv initiator în timpul unei faze de date la o operatie de citire, initiatorul trebuie sa activeze semnalul *PERR.* Initiatorul unei tranzactii are responsabilitatea de a raporta detectarea unei erori de paritate. În acest scop, initiatorul trebuie sa monitorizeze semnalul.

SERR (System Error)

Acest semnal poate fi activat de orice dispozitiv PCI pentru a raporta erori de paritate a adreselor, erori de paritate a datelor în timpul unui ciclu special, sau erori critice diferite de cele de paritate. Acest semnal este considerat ca ultima modalitate de raportare a erorilor serioase. Erorile care nu sunt catastrofale si cele corectabile trebuie semnalate în alte moduri. La calculatoarele PC, *SERR* determina în general o întrerupere NMI catre procesorul sistemului.

REQ (Request)

Este activat de initiator pentru a indica o cerere de magistrala. Aceasta linie este conectata la arbitrul de magistrala. Metoda de arbitrare a magistralei nu este descrisa în specificatiile PCI; se pot implementa diferite metode. Este specificat doar faptul ca arbitrul de magistrala trebuie sa utilizeze un algoritm prin care sa se evite blocajele; fiecarui dispozitiv *master* potential trebuie sa i se permita accesul la magistrala.

GNT (Grant)

Este activat de arbitrul de magistrala pentru a indica acordarea magistralei pentru initiator. Atunci când detecteaza acest semnal, initiatorul trebuie sa astepte terminarea tranzactiei în curs de catre initiatorul curent.

CLK (Clock)

Reprezinta semnalul de ceas utilizat pentru sincronizarea tuturor tranzactilor, inclusiv a arbitrajului de magistrala. Toti parametrii de sincronizare ai magistralei sunt specificati relativ la frontul crescator al semnalului de ceas. Frecventa semnalului de ceas se poate modifica în orice moment, cu condtia sa nu existe cereri de magistrala si semnalul *LOCK* sa nu fie activ. De asemenea, ceasul poate fi oprit în starea *low* (pentru reducerea puterii consumate).

RST (Reset)

Semnalele optionale

AD[63-32]

Reprezinta extensia la 64 de biti a magistralei de date *AD[31-0]*. Aceste linii nu sunt utilizate în timpul fazei de adresare a unui transfer (cu exceptia cazului în care se utilizeaza adresarea pe 64 de biti).

C/BE [7-4]

Reprezinta extensia semnalelor C/*BE[3-0]*. Se utilizeaza numai în timpul fazei de date a unui transfer (cu exceptia cazului în care se utilizeaza adresarea pe 64 de biti).

PAR64 (Parity for Upper Doubleword)

Este bitul de paritate paritate asociat *AD[63-32)* si *C/BE [7-4]*.

REQ64 (Request 64 bit transfer)

Semnal generat de initiatorul curent pentru a indica faptul ca doreste executarea transferurilor utilizând extensia la 64 de biti a liniilor de date. Semnalul *REQ* 64 are aceesi sincronizare ca si semnalul *FRAME.*

ACK64 (Acknowledge 64-bit transfer)

Semnal generat de tinta curent adresata (daca aceasta permite transferuri pe 64 de biti) ca raspuns la activarea semnalului *REQ64* de catre initiator. Semnalul *ACK*64 are aceeasi sincronizare ca si semnalul *DEVSEL*

LOCK

Acest semnal este utilizat de catre un initiator PCI care solicita accesul exclusiv la un dispozjtiv tinta de

memorie în timpul a doua sau mai multe tranzactii separate. Acest semnal este prevazut cu scopul de a permite operatii de citire/modificare/scriere la operatiile cu semafoare. Daca un dispozitiv PCI contine o memorie executabila sau o memorie care contine date de sistem (gestionate de sistemul de operare), trebuie sa implementeze aceasta functie de blocare a resursclor.

CLKRUN

Acest semnal este optional si este definit pentru sistemele portabile.

INTA, INTB, INTC, INTD

Acest semnal este o iesire de la puntea UCP/PCI si o intrare pentru subsistemele de memorie conectate la magistrala PCI care pot utiliza o memorie *cache.*

SBO (Snoop Back Off)

Semnalul *SBO* estre activat de puntea UCP/PCI pentru a indica faptul ca prin accesul curent la memorie urmeaza sa se citeasca sau sa se actualizeze informatii perimate din memorie. Semnalul *SBO* are semnificatie numai atunci când semnalul *SDONE* este de asemenea activat de catre puntea UCP/PCI. Atunci când ambele semnale sunt activate, subsistemul de memorie PCI curent adresat trebuie sa raspunda prin semnalarea unei retransmisii la initiatorul curent.

SDONE (Snoop Done)

Acest semnal este iesire de la puntea UCP/PCI si o intrare pentru subsistemele de memorie conectate la magistrala PCI care pot utiliza o memorie *cache.* Semnalul *SDONE* este dezactivat de puntea UCP/PCI în timp ce memoria *cache* a procesorului intervine într-o cerere dre acces la memorie executata de initiatorul curent. Puntea activeaza semnalul *SDONE* atunci când interventia s-a terminat. Rezultatele interventiei sunt indicate de catre semnalul *SBO.* Semnalul *SBO* dezactivat indica faptul ca initiatorul PCI acceseaza date valide din memorie si ca memoria poate accepta sau furniza datele respective. Semnalul *SBO* activat indica faptul ca initiatorol PCI acceseaza date perimate din memorie si nu trebuie sa termine accesul curent. Memoria va trebui sa termine accesul prin semnalarea faptului ca operatia trebuie repetata.

TCLK (Test Clock)

TDI (Test Data Input)

TDO (Test Data Out)

TMS (Test Mode Select)

TRST (Test Reset)

3.11.3. Arbitrarea magistralei PCI

Pentru a utiliza magistrala PCI, un dispozitiv trebuie, mai întâi, sa o câstige. Arbitrarea magistralei PCI utilizeaza un arbitru de magistrala, centralizat, dupa cu se vede în fig. 3.22. În cele mai multe variante de proiectari, arbitrul magistralei este continut într-unul din circuitele punte. Fiecare dispozitiv PCI are doua linii dedicate, care merg de la el la arbitru. O linie, REQ#; este utilizata pentru a cere magistrala. Cealalta linie, GNT#, este utilizata pentru a receptiona recunoasterea alocarii magistralei.

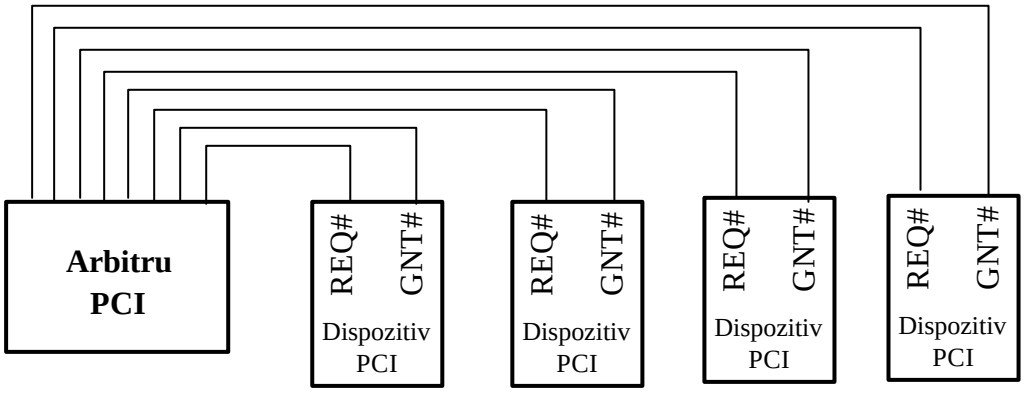

Figura 3.22. Magistrala PCI utilizata ca arbitru de magistrala, centralizat.

Pentru a cere magistrala, un dispozitiv PCI (inclusiv UCP), activeaza REQ# si asteapta pâna când vede ca linia GNT# este activata de arbitru. Când se întâmpla acest eveniment, dispozitivul poate utiliza magistrala pe durata ciclului urmator. Algoritmul utilizat de catre arbitru nu este definit de specificatia PCI. Sunt

permise arbitrarea "round robin", arbitrarea în functie de prioritati, precum si alte scheme. În mod clar, un arbitru trebuie sa fie corect, adica sa nu lase unele dispozitive sa astepte la nesfârsit.

Magistrala se aloca pentru o tranzactie, cu toate ca lungimea acestei tranzactii este. teoretic, nelimitata. Daca dispozitivul vrea sa realizeze o a doua tranzactie, sio nici un alt dispozitiv nu solicita magistrala, el o poate realiza, desi, în mod normal, între tranzactii trebuie inserat un ciclu inactiv.

Totusi în conditii speciale, în absenta unei competitii pentru magistrala, un dispozitiv poate realiza tranzactii una dupa alta, fara a trebui sa insereze un ciclu inactiv. Daca un master face un transfer foarte lung si alte dispozitive au cerul magistrala, arbitrul poate nega linia GNT#. Se presupune ca modulul master curent de pe magistrala urmareste linia GNT#, astfel încât atunci când sesizeaza negarea semnalului, trebuie sa elibereze magistrala pe ciclul urmator. Aceasta schema permite transferuri foarte lungi (care sunt eficiente) când este un singur candidat master pe magistrala, dar da, totusi, raspuns rapid pentru dispozitivele concurente.

Semnalul CLK comanda magistrala. Cele mai multe dintre celelalte semnale sunt sincrone cu acesta. Spre deosebire de magistrala ISA, o tranzactie pe magistrala PCI începe pe frontul descendent al semnalului CLK care este în mijlocul unui ciclu, în loc de începutul acestuia.

Cele 32 de semnale AD sunt pentru adresa si date (pentru tranzactiile pe 32 de biti). În general, pe durata ciclului 1 este activata adresa, iar pe durata ciclului 3 sunt activate datele. Semnalul PAR este un bit de paritate pentru AD. Semnalul C/BE# este utilizat pentru doua lucruri diferite. Pe ciclul 1, el contine comanda magistralei (citeste 1 cuvânt, citeste un bloc etc.). Pe ciclul 2 el contine o configuratie de 4 biti, ce specifica care octeti ai cuvântului de 32 de biti este valid. Utilizând C/BE# este posibil sa se citeasca sau scrie 1, 2, 3 octeti sau întreg cuvântul.

Semnalul FRAME# este activat de master-ul magistralei, pentru a începe o tranzactie pe magistrala. La o citire, de obicei, IRDY# este activat acelasi timp cu FRAME#. El spune ca master-ul este gata sa accepte datele de intrare. La o scriere, IRDY# este activat mai târziu, când datele sunt pe magistrala.

Semnalul IDSEL# este legat de faptul ca fiecare dispozitiv PCI trebuie sa aiba un spatiu de configuratie de 256 de biti, pe care alte dispozitive sa-l poata citi (prin activarea IDSEL). Acest spatiu de configuratie contine proprietatile dispozitivului. Caracteristica de "autoconfigurare" (Plug'n Play) a unor sisteme de operare utilizeaza spatiul de configuratie pentru a afla ce dispozitive sunt pe magistrala.

Am ajuns, acum, la semnalele activate de slave. Primul dintre acestea, DEVSEL#, anunta ca slave-ul a detectat adresa sa pe liniile AD si este pregatit sa se angajeze în tranzactie. Daca DEVSEL# nu este activat pe o anumita durata limitata de timp, master-ul presupune ca dispozitivul adresat este fie absent, fie stricat.

Cel de al doilea semnal de la slave este TRDY#, pe care slave-ul îl activeaza la citiri, pentru a anunta ca datele sunt pe liniile AD si la scrieri, pentru a anunta ca este pregatit sa accepte date.

Urmatoarele trei semnale sunt pentru raportarea de erori. Primul dintre acestea este STOP#, pe care slave-ul îl activeaza daca se întâmpla ceva dezastruos si vrea sa abandoneze tranzactia curenta. Urmatorul, PERR#, este utilizat pentru a raporta o eroare de paritate pentru date, pentru ciclul anterior. Pentru o citire, el este activat de master; pentru o scriere este activat de catre slave. Este sarcina receptorului sa faca actiunea corespunzatoare. În final, SERR# este pentru raportarea erorilor de adresa si a erorilor de sistem.

Semnalele REQ# si GNT# sunt pentru a efectua arbitrarea magistralei. Acestea nu sunt activate de master-ul curent de pe magistrala, ci mai exact de un dispozitiv ce doreste sa devina master pe magistrala. Ultimul semnal obligatoriu este RST#, utilizat pentru reinitializarea sistemului, daca utilizatorul apasa butonul RESET sau daca un dispozitiv sistem anunta o eroare fatala. Activarea acestui sistem initializeaza toate dispozitivele si reporneste calculatorul.

Am ajuns acum la semnalele optionale, cele mai multe dintre acestea fiind legate de extinderea de la 32 de biti la 64 de biti. Semnalele REQ64# si ACK64# permit master-ului sa ceara permisiunea pentru a realiza tranzactii pe 64 de biti si, respectiv, permite slave-ului sa le accepte. Semnalele AD, PAR63 si C/BE# sunt pur si simplu extensii ale semnalelor respective pentru 32 de biti.

Urmatoarele semnale nu au legatura cu diferenta de la 32 de biti la 64 de biti, ci cu sistemele multiprocesor, ceva ce placile PCI nu sunt obligate sa permita. Semnalul LOCK permite blocarea magistralei pentru tranzactii multiple. Urmatoarele doua sunt legate de ciclul de magistrala, pentru mentinerea coerentei memoriei intermediare.

Semnalele INTx sunt petru cereri de întrerupere. O placa PCI poate avea pâna la patru dispozitive logice, separate, pe ea, si fiecare poate avea propriile sale linii de cerere de întrerupere. Semnalele JTAG sunt pentru procedura de test IEEE 1149.1 JTAG. În sfârsit, semnalul M66EN este cablat fie la tensiunea de alimentare, fie la masa, pentru a stabili viteza ceasului. El nu trebuie schimbat pe durata functionarii sistemului.

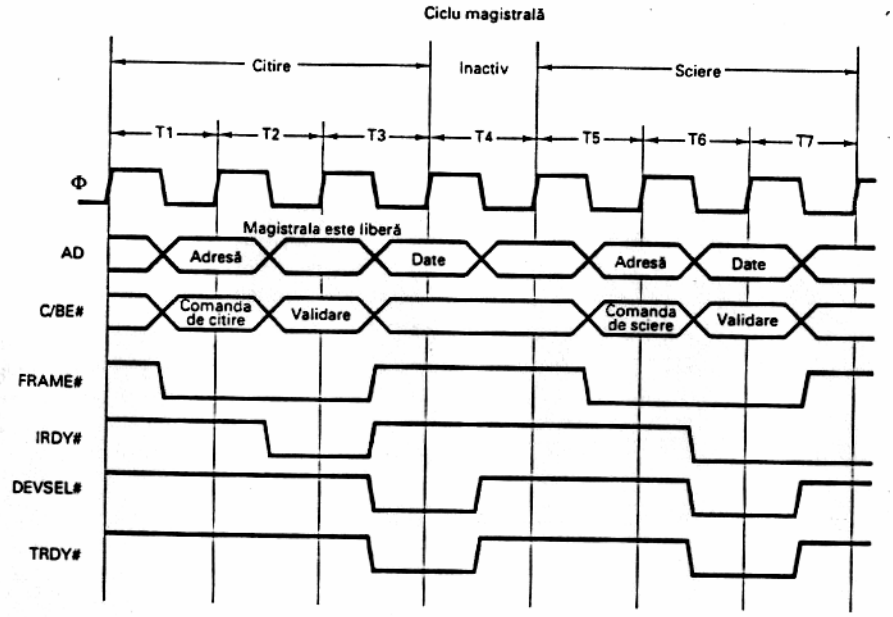

Fig. 3.23.

Magistrala PCI este foarte simpla (ca mod de functionare). Pentru a va da seama mai bine de aceasta, sa consideram diagrama din fig. 3.23. Aici se poate vedea o tranzactie de citire, urmata de un ciclu inactiv, urmat de o tranzactie de scriere pentru acelasi master de pe magistrala.

Pe frontul descrescator al ceasului pe durata lui T_1, master-ul pune adresa de memorie pe AD si comanda pentru magistrala pe C/BE#. El activeaza, apoi, FRAME# pentru a începe tranzactia pe magistrala.

Pe durata T_2 master-ul lasa magistrala de adrese libera (adica în starea de mare impedanta), în pregatirea ei pentru slave, care o va controla pe durata T_3. De asemenea, master-ul modifica C/BE# pentru a arata ce octeti din cuvântul adresat vrea sa-i valideze (adica sa-i citeasca).

În T_3 slave-ul activeaza DEVSEL#, astfel ca master-ul stie ca acesta a luat adresa si intentioneaza sa raspunda. De asemenea, el pune datele pe liniile AD si activeaza TRDY# pentru a anunta master-ul. Daca slave-ul nu este capabil sa raspunda asa de repede, el va trebui sa activeze totusi DEVSEL#, pentru a anunta prezenta sa, dar mentine TRDY# negat pâna când poate sa preia datele.

Aceasta procedura poate introduce una sau mai multe stari de asteptare (wait).

În acest exemplu (si adesea în realitate), urmatorul ciclu este inactiv. Începând din T_5 vedem acelasi master care initiaza o scriere. El începe, ca de obicei, prin plasarea adresei si a comenzii pe magistrala. Numai acum, în cel de-al doilea ciclu, el activeaza datele. Întrucât acelasi dispozitiv controleaza liniile AD, nu este necesar un ciclu de eliberare a magistralei. Pe T_7, memoria accepta datele.

3.12. Magistrala seriala universala (USB)

Magistrala PCI este adecvata pentru atasarea perifericelor de mare viteza la un calculator, dar este prea costisitor sa existe o interfata PCI pentru fiecare dispozitiv I/E de viteza scazuta, cum ar fi o tastatura sau un mouse. Istoric, fiecare dispozitiv de I/E standard a fost conectat la calculator într-un mod specific, lasând libere câteva conectoare ISA sau PCI pentru a adauga noi dispozitive. Din pacate, aceasta schema a fost însotita de probleme, chiar de la început. De exemplu, fiecare nou dispozitiv de I/E vine, adesea, cu propria sa placa ISA sau PCI. Utilizatorul este, adesea, responsabil pentru pozitionarea comutatoarelor si a unor conexiuni (jumpere) pe placa, si asigurarea ca aceste pozitionari nu vin în conflict cu alte placi. Apoi utilizatorul trebuie sa desfaca cutia (calculatorului), sa insereze cu grija placa, sa închida cutia si sa reporneasca calculatorul. Pentru multi utilizatori, acest proces este dificil si cu mari sanse se eroare. În plus, numarul de conectori ISA si PCI este strict limitat (de obicei doi sau trei). Placile cu autoconfigurare (Plug'n Play) elimina pozitionarea unor conexiuni (jumpere), dar utilizatorul trebuie sa deschida, totusi, calculatorul si sa introduca placa, iar numarul de conectori de magistrala este si el limitat.

Pentru a rezolva aceasta problema pe la mijlocul anilor 1990, reprezentanti de la sapte companii (Compaq, DEC, IBM, Intel, Microsoft, NEC si Northen Telecom) s-au strâns pentru a proiecta un mod mai bun de a atasa dispozitivele I/E, de viteza redusa, la un calculator. De atunci sute de alte companii li s-au alaturat.

Standardul rezultat este denumit USB (Universal Serial Bus, rom: magistrala seriala universala) si este implementat, pe scara larga, în calculatoarele personale. El este descris mai detaliat în Anderson, (1997) si Tan (1997).

Câteva dintre obiectivele companiilor care au conceput initial USB-ul si care au început proiectul, au fost urmatoarele:

1. Utilizatorii nu trebuie sa aiba de pozitionat comutatoare sau conexiuni (jumpere) pe placi sau pe dispozitive.
2. Pentru a instala noi dispozitive de I/E utilizatorii nu trebuie sa deschida cutia.
3. Trebuie sa existe un singur tip de cablu, bun pentru conectarea tuturor dispozitivelor.
4. Dispozitivele de I/E trebuie sa se alimenteze singure, prin cablu.
5. Se pot atasa pâna la 127 de dispozitive la un singur calculator.
6. Sistemul ar trebuie sa suporte dispozitivele în timp real (de exemplu, sunet, telefon).
7. Dispozitivele trebuie sa poata sa fie instalate în timp ce calculatorul functioneaza.
8. Dupa instalarea unui nou dispozitiv sa nu trebuiasca repornit calculatorul.
9. Noua magistrala si dispozitivele ei de I/E ar trebui sa fie.

USB îndeplineste toate aceste conditii. Ea este proiectata pentru dispozitive de viteza redusa cum ar fi tastaturi, mouse-uri, camere de luat vederi, scanere, telefoane digitale si altele. Banda de trecere totala pentru USD este de 1.5 Mo/sec., care este suficienta pentru un numar substantial de astfel de dispozitive. Aceasta limita scazuta a fost aleasa pentru a mentine un cost scazut.

Un sistem USB consta dintr-un hub central (root hub) care se conecteaza la magistrala principala (vezi fig. 3.20). Acest hub are socluri pentru cabluri ce se pot conecta la dispozitive de I/E sau la alte hub-uri de extindere, pentru a furniza mai multe socluri, astfel ca topologia unui sistem USB este un arbore cu radacina sa la hub-ul central, în interiorul calculatorului. Cablurile au diferiti conectori, la capatul hub-ului si la capatul dispozitivului, pentru a preveni interconectarea accidentala a doua socluri cu hub-uri.

Cablul consta din patru fire: doua pentru date, unul pentru alimentare (+5 volti) si unul pentru masa. Sistemul de semnalare transmite un 0 ca o tranzitie a tensiunii si un 1 ca absenta unei tranzactii de tensiune, astfel ca atât timp cât dureaza 0-urile se genereaza un flux de pulsuri.

Când se conecteaza un nou dispozitiv de I/E, hub-ul radacina detecteaza acest eveniment si întrerupe sistemul de operare. Apoi sistemul de operare interogheaza dispozitivul, pentru a determina ce este si cât din banda USB necesita. Daca sistemul de operare decide ca banda de trecere este suficienta pentru dispozitiv, el asociaza noului dispozitiv o adresa unica (1-127) si încarca aceasta adresa si alte informatii în registrele de configurare din dispozitiv. În acest mod, noi dispozitive pot fi adaugate imediat, fara a fi necesara nici o informatie de configurare din partea utilizatorului si fara a trebui sa se instaleze noi placi ISA sau PCI. Placile neinitializate pornesc de la adresa 0, astfel ca ele pot fi adresate. Pentru a face cablarea mai usoara, multe dispozitive USB contin hub-uri incorporate pentru a accepta dispozitive suplimentare USB. De exemplu, un monitor poate sa aiba doua socluri de hub pentru a amplasa difuzoarele stânga dreapta.

Din punct de vedere logic, sistemul USB poate fi vazut ca un set de canale de biti, de la hub-ul central la dispozitivele de I/E. Fiecare dispozitiv împarte canalul sau în cel mult 16 subcanale, pentru diferite tipuri de date (de exemplu, audio si video). Pe fiecare canal sau subcanal, datele merg de la hub-ul central la dispozitiv sau invers. Nu exista trafic între doua dispozitive de I/E.

La exact fiecare 1.00±0.05 msec., hub-ul emite un nou cadru pentru a mentine toate dispozitivele sincronizate în timp. Un cadru este asociat ca un canal de biti si consta din pachete, primul fiind de la hub-ul radacina catre dispozitiv. Urmatoarele pachete din cadru pot, de asemenea, sa fie în aceasta directie sau pot merge înapoi de la dispozitiv la hub-ul central. În fig. 3-54 este prezentata o secventa de patru cadre.

În fig. 324. nu se efectueaza nimic în cadrele 0 si 2, astfel încât este necesar doar un pachet SOF (Start of Frame), de început de cadru. Acest pachet este transmis, întotdeauna, la toate dispozitivele. Cadrul 1 este o interogare, de exemplu o cerere pentru ca un dispozitiv de scanare sa returneze bitii pe care i-a gasit în imaginea pe care o scaneaza. Cadrul 3 consta din datele ce se transmit la un dispozitiv, de exemplu la o imprimanta.

USB suporta patru tipuri de cadre: control, izocron, transfer cantitati mari de date (bulk) si întrerupere. Cadrele de control sunt utilizate pentru a configura dispozitive, a le transmite comenzi si a interoga despre starea lor. Cadrele izocrone sunt dispozitive în timp real cum ar fi microfoane, difuzoare si telefoane care au nevoie sa transmita sau sa accepte date la intervale de timp precise. Ele au o întârziere predictibila, dar nu permit retransmisia în cazul unor erori. Cadrele de transfer cantitati mari de date sunt pentru transferuri mari la sau de la dispozitive fara cerinte de timp real, cum ar fi imprimantele. În sfârsit, cadrele întrerupere sunt necesare deoarece USD nu suporta întreruperi. De exemplu, în loc sa avem o tastatura care sa genereze o întrerupere ori de câte ori se apasa o tasta sistemul de operare o poate interoga la fiecare 50 msec. pentru a prelua tastele apasate pâna atunci.

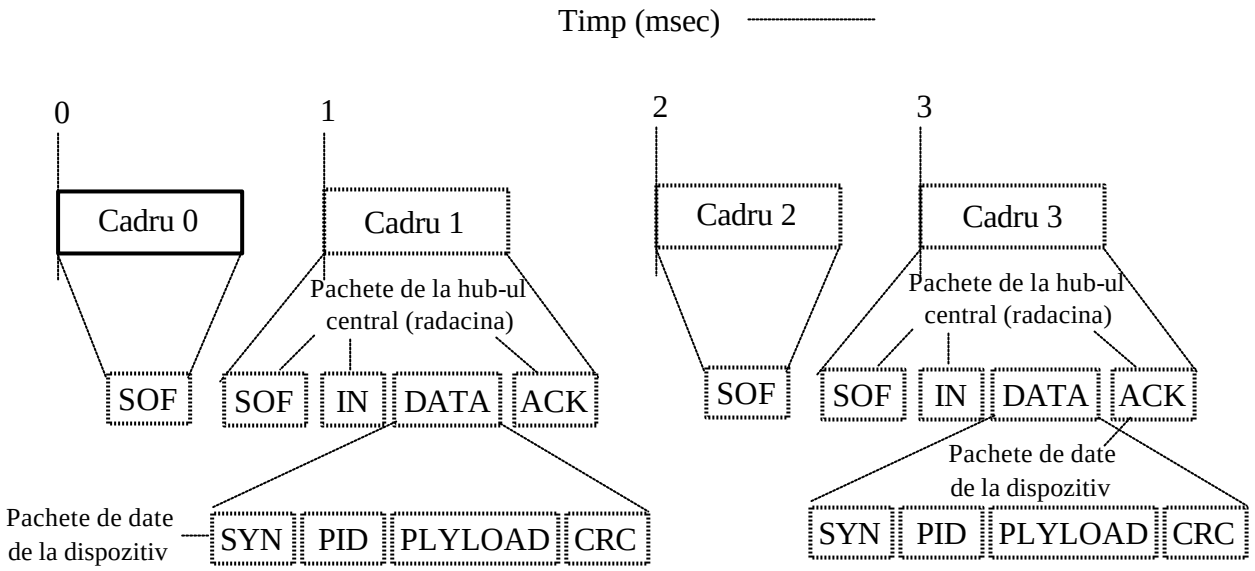

Fig. 3.24. Hub-ul radacina USB transmite cadre la fiecare 1.00 msec.

Un cadru consta din unul sau mai multe pachete, câteva, posibil, în fiecare directie. Exista patru tipuri de pachete: control sistem (token), date, negociere si special. Pachetele token sunt de la radacina la root la un dispozitiv si sunt pentru controlul sistemului. Pachetele SOF, IN si OUT din fig. 3.24. sunt pachete de control sistem (token). Pachetul SOF (Start of Frame) este primul în fiecare cadru si marcheaza începutul cadrului. Daca nu este nimic de facut acolo, pachetul SOF este singurul din cadru. Pachetul de control sistem IN este o interogare, care cere dispozitivului sa obtina anumite date. Câmpurile din pachetul IN specifica canalul de biti interogat, astfel ca dispozitivul stie care sunt datele de obtinut (fdaca canalul are mai multe fluxuri). Pachetul de control sistem OUT anunta ca vor urma datelke pentru dispozitiv. Cel de al patrulea tip de pachet pentru controlul sistemului SETUP (care nu este prezentat în figura), este utilizat pentru configurare.

Pe lânga pachetul de control sistem (token), mai exista alte trei tipuri. Acestea sunt pachetele: DATA utilizat pentru transmisie pâna la 64 de octeti de informatie în orice directie, negociere si pachete speciale. Formatul unui pachet de date este prezentat în fig. 3.24. El consta dintr-un câmp de sincronizare de 8 biti, un tip de pachet de 8 biti (PID), sarcina, adica datele de transmis si CRC-ul de 16 biti (Cyclic Redundancy Code;rom: cod redundant ciclic), pentru a detecta erori. Sunt definite trei tipuri de pachete de negociere: ACK (pachetul anterior de date a fost bine receptionat), NAK (s-a detectat o eroare CRC) si STALL (va rog asteptati - acum sunt ocupat).

Sa examinam acum, din nou, Fig. 3.24. La fiecare 1.00 msec trebuie transmis un cadru de la hub-ul radacina (central), chiar daca nu este nimic de facut. Cadrele 0 si 2 constau doar dintr-un pachet SOF, indicând ca nu s-a facut nimic. Cadrul 1 este o inerogare, astfel ca el începe cu transmiterea pachetelor SOF si IN de la calculator la dispozitivele de I/E, urmate de un pachet DATA de la dispozitiv la calculator. Pachetul ACK spune dispozitivului ca datele au fost receptionate corect. În cazul unei erori catre dispozitiv se va transmite un NAK si pachetul se va retransmite în cazul transferurilor de cantitati mari de date (dar nu si cel al datelor izocrone). Cadrul 3 este similar, ca structura, cu cadrul 1, cu exceptia faptului ca , acum, fluxul de date este de la calculator la dispozitiv.